uc3m | Universidad Carlos III de Madrid

UNITS 7 AND 8: SEMANTIC ANALYSIS and ERROR HANDLING

We want to incorporate a repetitive sentence into a high-level language. The sentence can be represented by the following regular expression:

repeat (identifier | number) >> sentence^{+} <<

A program consists of at least one statement, where statements can be assignments, conditionals, and loops.

NOTE: The symbols "|" and "+" are part of the regular expressions, the others are part of the language.

It is required:

1. Define the grammar G that would generate valid programs of this programming language. Consider the assignment and conditional statements as terminal symbols of the grammar.

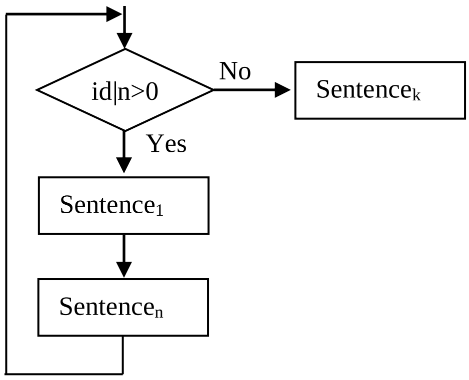
2. Describe the semantic routines of the grammar G that generate intermediate code in quartets with the following instructions, where *pos* are memory addresses, registers, or a number, and *reg*, *reg1* and *reg2* can be a record or a number. Write the semantic routines for the two interpretations that can be made about the execution flow of the loop:

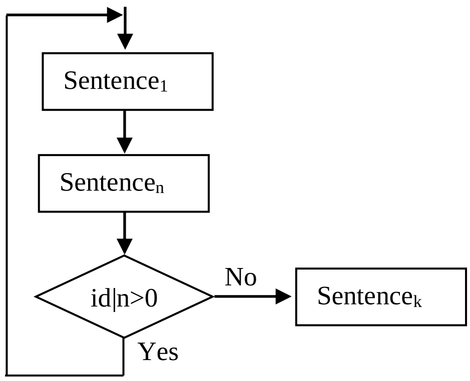
repeat (id | n) >>
sentence$_1$
…
sentence$_n$
<<
sentence$_k$

David Griol Barres, Antonio Berlanga de Jesús, Jesús García Herrero, Juan Manuel Alonso Weber

Mode A

Mode B





Statement	**Meaning**
(move, pos_1,, pos_2)	$pos_2 \leftarrow pos_1$
(push, pos_1, ,)	includes the contents of pos_1 into the stack
(pop, , , pos_1)	$pos_1 \leftarrow$ top of the stack
(label, , , *label*)	defines a label
(goto, , , *label*)	go to a label
(return, , , *reg*)	go to the address specified by *reg*
(if, *reg*, , *label*)	go to label *reg* es -1
(<, *reg*, , *label*)	go to label if the contents of *reg* is lower or equal to 0
(+, reg_1, reg_2, *reg*)	$reg \leftarrow reg_1 + reg_2$
(-, reg_1, reg_2, *reg*)	$reg \leftarrow reg_1 - reg_2$
(*, reg_1, reg_2, *reg*)	$reg \leftarrow reg_1 * reg_2$
(/, reg_1, reg_2, *reg*)	$reg \leftarrow reg_1 / reg_2$